

アジャイルソフトウェア開発 PBL における 形成的アセスメントの効果測定

日戸 直紘 伊藤 恵 大場 みち子

多くの情報系大学において、Project-Based Learning(PBL) が広く取り入れられ、その有効性や実践例が多数報告されている。ソフトウェア開発を行う PBL では、Scrum に代表されるアジャイル開発方法論を採用する大学が増加している。Scrum では、透明性・検査・適応の実践が求められるが、開発・マネジメント経験の少ない学生が透明性の確立、プロジェクトを検査・適応する難易度は高い。また、講義設計や学びの状況、質を改善するには、学習途中でそこまでの成果や達成度を把握し、学習を促す形成的な評価・アセスメントを行うのが望ましい。しかし、教育者が PBL 専任でない場合や、学生が時間外にも活動するなど、形成的な評価を行うには多くの課題がある。これらの課題解決を目指し、プロセスを評価し改善するモデルである能力成熟度モデル統合 (CMMI) に着目した。本稿では、筆者らが提案する CMMI に基づいた、プロジェクト状況を検査可能な評価手法に対し、PBL における形成的アセスメント効果の検証について報告する。

Project-Based Learning (PBL) has been widely adopted in many information-oriented universities and its effectiveness and practical examples have been reported. In PBL that develops software, there are an increasing number of universities that adopt the Agile development methodology represented by Scrum. Although Scrum requires the practice of transparency, inspection, and adaptation, it is highly difficult for students with a little development and management experience to establish transparency and inspect and adapt projects. In addition in order to improve the situation and quality of lecture design and learning it is desirable to grasp the results and achievement so far during learning and to carry out formative evaluation and assessment that encourage learning. However, there are many problems in performing formative evaluation such as when each educator is not full-time PBL or overtime activities. In order to solve these problems, we focused on the Capability Maturity Model Integration (CMMI), which is a model to evaluate and improve the software process. In this paper we report the effectiveness of formative assessment in PBL using the evaluation method that can inspect our proposed the project situation based on CMMI.

1 はじめに

文部科学省は、IT 人材の不足が今後一層深刻化する可能性が高いなどから、高度 IT 人材の育成を日本の極めて重要な課題としている [1]。高等教育機関において高度 IT 人材の育成が求められる背景のなかで、Project-Based Learning(PBL) が注目されている。PBL とは課題解決型学習を指し、学習者が課題

解決に向け主体的 (あるいは能動的) に取り組む実践的な授業法である。座学等で学んだ体系的知識の定着と利活用への効果も期待されており、より良い教授法を目指し試行錯誤が繰り返されている。

ソフトウェア開発を目的とするソフトウェア開発 PBL では、社会のニーズや課題、IoT などの要素技術を掛け合わせ、学生がアイデア・実装・検証を繰り返し、成果物を生み出し、成果報告会を行うといった流れが一般的である。そのソフトウェア開発 PBL に、アジャイル開発方法論を採用する大学が増加している (以下、アジャイルソフトウェア開発 PBL)。筆者らの調査によると、アジャイルソフトウェア開発 PBL を実施している大学のうち、9 割が Scrum [2] を導入していることがわかった。

Effects of Formative Assessment in Agile Software Development PBL

Naohiro Hinoto, 公立はこだて未来大学 システム情報科学研究科, Graduate School of Systems Information Science, Future University Hakodate.

Kei Ito and Michiko Oba, 公立はこだて未来大学, Future University Hakodate.

Scrum はプロダクトを開発・提供・保守するためのフレームワークであり、可能な限り価値の高いプロダクトを生産的かつ創造的に届けるためのものである[2]。Scrum において、透明性・検査・適応の 3 本柱は非常に重要な軸となっており、経験的プロセス制御の実現を支えている。経験的プロセス制御とは、経験主義とも言い、実際の経験と既知に基づく判断によって知識が獲得できるというものである[2]。そのため、経験や知識が乏しい学生自身が主体となって学ぶ PBL において、Scrum の導入、3 本柱の実現は難易度が高い側面を持ち合わせている。加えて、アジャイル開発経験の無い教員が多いことや、アジャイル開発を教える講義が無いなどからコーチングが難しいといった教育者側の課題も考えられる。

実践的な教育を実施する情報技術人材育成のための実践教育ネットワーク形成事業 (通称: enPiT) [3] では、これらの課題をフォローアップする取り組みを行っている大学も存在する。enPiT の一部大学では、アジャイル開発導入のコーチングを行っているアジャイルコーチ[4]などの企業人らと協力し、独自のカリキュラムや合宿を設計している。しかし、PBL チーム数の多さや予算などのコスト面における課題などから、一部の高等教育機関のみの実施に留まっており、実施している enPiT 参画大学も enPiT の予算次第では実施が難しくなる可能性が高いのが現状である。

PBL の学習評価、特にプロセス (活動そのもの) の評価を適切に実施することは困難であると多数報告されており[5][6][7]、PBL の学習評価は PBL 運営の一つの課題となっている。PBL の担当教員は、専任ではなく通常の教員業務との兼任の場合が多く、学生は講義時間外に活動することも多いため、常に学習者を観測することは難しい。先のアジャイルソフトウェア開発 PBL の課題から、講義改善のための評価 (形成的評価) を定量的に行うのは非常に重要であるが、PBL 評価の課題なども踏まえて考えると実施自体の困難性や教育者の負担が大きいなどの課題も重なり実現が難しい。

本研究では、アジャイルソフトウェア開発 PBL を実践する学生の学び、Scrum 導入の支援および講義指導、カリキュラム改善などの複数の課題解決を目的

とした定量的な評価手法を提案する。

PBL 実施中の学生が行うプロセスの継続的な評価方法が求められることから、筆者らは、Cability Maturity Model Integration (CMMI) [8][9] に着目し、CMMI のいくつかの領域と独自に開発した領域を用いた評価手法を開発した。

本稿では以降、2 章で公立はこだて未来大学における複数の PBL に関する特徴について述べ、3 章では CMMI について概説する。4 章では筆者らが提案する評価手法について述べ、5 章で PBL での実験について述べる。6 章では Communication Tool の可視化について述べ、7 章では実践結果について分析・考察していく。8 章では本研究の結論と今後の課題について述べ、まとめる。

2 公立はこだて未来大 PBL の特徴

2 章では、公立はこだて未来大学 (以下、未来大) における複数の PBL に関する特徴について述べる。

未来大では、開学初期から講義カリキュラムに PBL を採り入れ、コース・分野を超え、様々な内容の PBL を実践してきた。学部 3 年次に開講される全学科必修の PBL であるシステム情報科学実習の他に、全学年参加可能な高度 ICT 演習と呼ばれる単位認定の無い自由参加型の PBL を行っている。

以降、システム情報科学実習と高度 ICT 演習の概要について解説する。

2.1 システム情報科学実習の概要

システム情報科学実習 (通称: プロジェクト学習) は、未来大の学部 3 年次の全コースに開講されている必修通年型の PBL である。本 PBL では、情報システムコース・高度 ICT コース・情報デザインコース・知能システムコース・複雑系コースの垣根を超えたコース混合の学生によって、プロジェクトを構成されている。毎年 20~30 プロジェクトが開講され、1 プロジェクトあたり 10~15 名程度の学生が参加する。毎週水・金曜日の 4・5 限 (14:50~18:00) が講義時間に設定されており、ソフトウェアだけでなくハードウェアやワークショップ等のイベント企画など様々なテーマに 1 年間取り組む。ゴールとして、成果報

告書の作成や学内向け成果発表会のほか、企業や自治体向けに成果を報告・発表するプロジェクトも多く、未来大の特徴的な演習科目となっている。

2.2 高度 ICT 演習の概要

高度 ICT 演習は、実社会の課題やニーズを理解し、問題解決につなげるシステムの提案から実装までを行う単位認定のない自由参加型の演習である。本演習は、学部生・大学院生を交えた学年・コースの壁がない学年混成型 PBL であり、学部 1 年生から修士 2 年生まで様々な学年・コースの学生がプロジェクトを組み活動している。演習のテーマは、学生が主体となり学生自らがテーマ提案を行っており、年度を超えた継続テーマに取り組むプロジェクト・学生も多い。

3 CMMI の概要

3 章では、筆者らが着目した CMMI(Capability Maturity Model Integration) について概説し、本研究で扱うプロセス領域について述べる。その後、CMMI における評価について述べていく。

企業におけるソフトウェア開発では、品質・生産性の向上、工期短縮を目的とした、プロセス評価・改善活動が盛んに行われている。プロセス評価・改善活動とは、組織やプロジェクトの有効性を評価し、様々な情報を総合的に分析することでプロセス改善に有効な策を展開していく活動のことである。プロセス評価を行うための業界標準とされる CMMI は、米国・中国・インドを中心に世界の多くの企業が採用している。

以下、CMMI について述べていく。

3.1 Capability Maturity Model Integration

CMMI の前身である CMM(Capability Maturity Model) は、1989 年にソフトウェア開発の企業を客観的に評価できる指標として、米国国防総省とカーネギーメロン大学 (CMU) により設立されたソフトウェア工学研究所 (SEI) によって誕生した。現在は、複数の改善モデルや知見を加えながら、能力成熟度モデル統合 (CMMI) となった。CMMI は、現在も CMU/SEI で開発・管理されており、2019 年 8 月現在の最新バージョンは、CMMI-V2.0 である [10]。しかし、翻訳活

動や企業への導入が進んでいないことから、事実上のデファクトスタンダードとなっているのは V1.3 である。そのため本研究では、CMMI-V1.3 を利用する。

3.2 プロセス領域

CMMI には、各活動内容に応じて分類されたプロセス領域と呼ばれる分類が存在し、CMMI-V1.3 では 22 のプロセス領域が定義されている。22 のプロセス領域は 4 つの「区分」に分けられており、重複はない。「区分」は、組織および組織内のプロジェクトが実施する活動内容を「プロセス管理」・「プロジェクト管理」・「エンジニアリング」・「支援」の 4 つに分類したものを示している。

22 のプロセス領域の中には、リスク分析やプロジェクトのコスト (金銭等) 管理など、PBL には不要もしくは適さない領域も存在する。そのため、CMMI を評価手法に活用する場合には、プロジェクトに応じて適切な領域を選択する必要がある。

3.3 CMMI における評価

CMMI を用いて企業やプロジェクトを評価する際は、CMMI アプレイザーと呼ばれる評価を行う専門家 (もしくはチーム) が評価を行う。アプレイザーは、外部かつ認定を受けた者が行うことで、評価への信頼性や妥当性、客観性を担保している。

4 効果測定手法の概要

4 章では、3 章で概説した CMMI に基づいた PBL のアセスメント効果を測定するための手法について述べていく。

4.1 対象領域と独自開発領域

3 章で解説したように、CMMI はプロジェクトの管理活動から実装、テスト、トレーニング (教育) など幅広い活動をカバーしている。そのため、学生が主体的に学ぶ PBL に適さない領域が複数あると考えられる。

本提案手法では CMMI の 5 領域と独自に開発した 1 領域の計 6 領域を選定・開発し、評価を行っていく。選定した 5 領域は、プロジェクトに必要なデータ収

集・分析、定期的なスケジュール等の計画・確認、プロセス実施の評価などの領域を中心に選定した。

表1は、これら6領域のCMMIにおける区分、プロセス領域名、選定理由についてまとめたものである。以降、各領域を表1に記載された領域A～Fで表記する。

表1 対象領域と独自領域の名称と選定理由

領域	区分	プロセス領域名	選定理由(概要)
A	プロジェクト管理	プロジェクト計画策定	学生が課題とするスケジュールなどを含む プロジェクト計画 を扱うため
B		プロジェクトの監視と制御	プロジェクト状況 を適切な頻度で 把握・是正処置するための領域 であるため
C	支援	測定と分析	あらゆるデータの 収集・分析・測定 を扱う領域であるため
D		プロセスと成果物の品質保証	プロセスや作業成果物などを客観的に評価 あるいは見直しを立てる領域であるため
E	レビュー/検証	検証	成果物に対するレビュー準備・実行・是正 スクラムイベントとの親和性が高いため
F	-	Scrum遵守(独自領域)	Scrum Guide に記載されたイベントやルール等の遵守に関して検査するため

CMMIはAgile/Scrumとの親和性は高い一方で、各領域にそれら进行评估する要素が分散していることや、Scrumのルール(役割)・イベントなどの直接的な項目は存在しない。そこで、“Scrum遵守”の領域としてScrum Guide[2]に記載されたルール・イベント等の遵守に関する独自領域を開発した。

4.2 ルーブリックの開発

本来のCMMIはアプレイザーが評価を行うのに対し、本手法では学生が主体的に学ぶというPBLの観点から、学生自身が評価を行うことを想定している。学生が自分たちの力で評価/改善を行うことで、プロセス改善・アセスメントの過程も学びに繋げていけると考えたためである。

学生が自己評価を行う課題として、主に以下が挙げられる。

- 客観性 客観的に自分たちを分析/評価できない
 - 妥当性 評価結果に理由/根拠がない(過大もしくは過小評価)
 - 知識 CMMIに関する知識を持っていない
- これらの課題から、CMMIの知識がない学生でも

評価可能であり、できる限り客観性/妥当性を持たせるため、各プロセス領域/評価項目に対して、ルーブリックを開発する。

ルーブリックを活用して、PBL受講者(学生)が一定期間(Scrumではスプリント)ごとに評価を行うことで、図1のように、学びの形成過程や課題箇所、成長などを自己評価できると考えた。

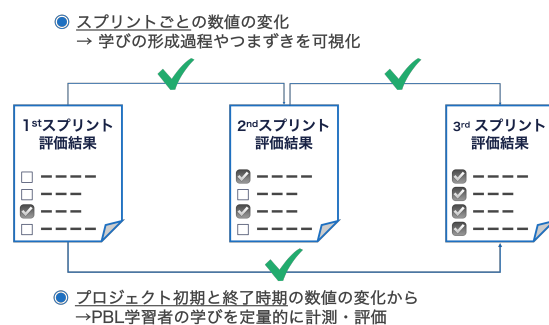


図1 学びの計測・評価

ルーブリックの評価項目は、CMMIの各プロセス領域に存在する項目に準拠し、評価基準において専門的な単語や表現をできる限り避けることで、CMMIの知識がない学生でも評価可能なルーブリックを目指した。

また、評価点は4段階評価(Lv.0～Lv.3)とし、評価結果をわかりやすくするため、以下の通り、%へ置き換えることとした。

- Lv.0 : 0%
- Lv.1 : 33%
- Lv.2 : 66%
- Lv.3 : 100%

各プロセス領域には、複数の評価項目が存在する。全プロセス領域合計で50項目以上存在するため、分析や評価を行いやすいようにプロセス領域ごとの評価を算出する必要がある。そこで、プロセス領域に含まれる評価項目の%の相加平均をそのプロセス領域の評価結果(0～100%)とすることで、評価結果・改善効果がわかりやすいように定義・算出した。

ルーブリックは可能な限り、客観性を保持する評

価方法であるが、評価者が学生のため客観性や妥当性には少し疑問が残る。そこで、日々の活動を可視化することにより、客観的な情報からも評価していくこととした。なお本稿では、PBL で利用されている Communication Tool を対象に日々の活動(プロセス)を捉えていく。

5 PBL における実験

本評価手法を用いることでプロジェクトを評価可能か検証するため、進行中の PBL を対象に実験を行った。

各スプリントのふりかえり(スプリントレトロスペクティブ)に加えて、提案手法による評価を実施してもらった。手順は以下の通りである。

1. ルーブリックが記載された評価シート
(Google Spreadsheet)を参照しながらの評価
2. 各評価項目に対し、自分たちがどのレベルに当てはまるのかルーブリックを基に議論
3. 当てはまるレベルを Lv.0~3 から選択
4. 手順 2~3 を全評価項目分実施
5. 必要に応じて評価結果を振り返りに活用

これらの手順をスプリントごとに1度、チームごとに実施してもらうことで、プロセス/形成的アセスメントの評価を行っていく。

5.1 実験対象プロジェクト

実験対象は、2 プロジェクト 18 チーム、計 50 名とし、プロジェクト活動期間のうち開発期間である 2018 年 10 月~12 月の 3ヶ月間実施した。

5.1.1 ミライケータイプロジェクト

ミライケータイプロジェクトは、システム情報科学実習の1プロジェクトであり、数年後当たり前となっているサービスを想定したアプリケーション開発を通して、企画から開発、ビジネスモデルを企業へ提案をする実践的な開発プロジェクトである[11]。毎年新たなサービスを企画・開発しており、その規模は連携大学もあわせ 40 名前後にも及び、PBL としては、大規模なプロジェクトである。

2018 年度は、学部 3 年生計 38 名がプロジェクトに参画しており、Scrum 手法を取り入れ、プロジェ

クトを進めている。プロジェクト内に 3 サービスが設置されており、さらに各サービスが 4 開発チームにわかれているため、計 12 チーム (4 開発チーム× 3 サービス) を実験対象とした。

5.1.2 FinTech プロジェクト

FinTech プロジェクトは、高度 ICT 演習にて 2017 年度発足された、学内向け仮想通貨を企画・開発するプロジェクトである。プロジェクトは、R&D チームと応用開発チームの 2 チームで構成されている。R&D チームはブロックチェーンを含めた仮想通貨そのものを開発、応用開発チームは仮想通貨を利用する際の取引所の様な役割を果たす仮想通貨アプリを開発している。本実験では、Scrum を取り入れている応用開発チームを実験対象としている。応用開発チームはさらに 6 チームに細分化されており、全 6 チーム計 12 名を対象とした。学部 1 年生 2 名、学部 2 年生 5 名、学部 3 年生 5 名が混成しており、多学年 PBL となっている。

5.2 評価結果

システム情報科学実習のミライケータイプロジェクト、高度 ICT 演習の FinTech プロジェクトにおいて実施した提案手法による評価結果を各領域/スプリントごとに示す。

5.2.1 ミライケータイプロジェクト

チーム数が多い関係上、システム情報科学実習の評価結果は、4 チームの平均値を各サービスの評価結果として、Service1~3 に示した。1 スプリント/2 週間であることや、開発期間/プロジェクト計画上の理由から、3 スプリント分のみの計測となった(図 2~図 7)。各図の縦軸は評価の点数 (%) スプリント、横軸はスプリントを表している。

5.2.2 高度 ICT 演習

6 開発チームの評価結果を Team A ~ Team F として示した。FinTech プロジェクトでは、1 スプリント/1 週間のため、システム情報科学実習と同様の期間である 6 スプリント分の計測を行った(図 8~図 13)。各図の縦軸は評価の点数 (%) スプリント、横軸はスプリントを表している。

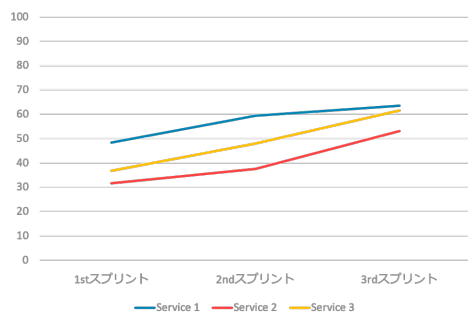


図 2 “scrum 遵守”領域の評価結果

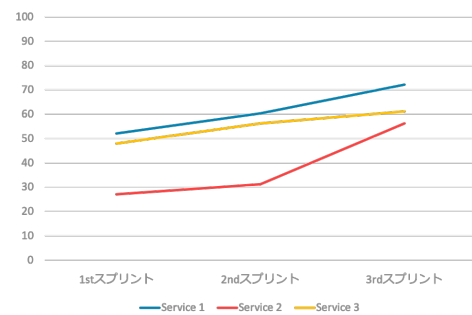


図 5 “検証”領域の評価結果

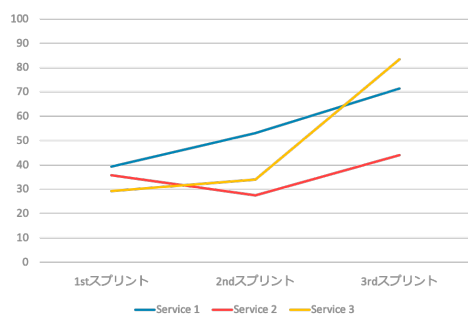


図 3 “プロジェクト計画策定”領域の評価結果

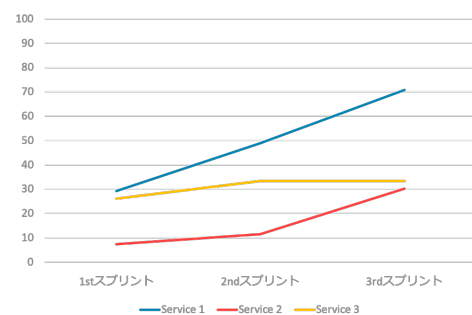


図 6 “測定と分析”領域の評価結果

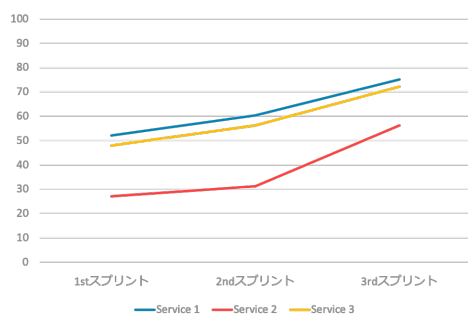


図 4 “プロセスと成果物の品質保証”領域の評価結果

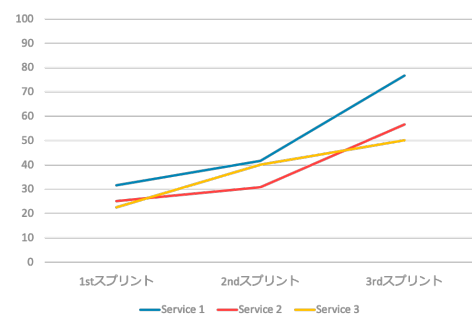


図 7 “プロジェクトの監視と制御”領域の評価結果

6 Communication Tool の可視化

PBL で利用されている Communication Tool の log を可視化することによって、評価データをより客観的かつ詳細に分析できると考えた。

実験対象の 2 プロジェクトで最も活発に利用されていた Slack[12] を可視化の対象とした。Slack とは、

主にチーム活動で用いられる Communication Tool でビジネスに必要な機能が多数備わっており、多くの企業や PBL で利用されている。

6.1 可視化システム

可視化ツールには、Microsoft 社の Power BI[13] を使用し、CData 社の Rest Driver[14] を用いた。Slack

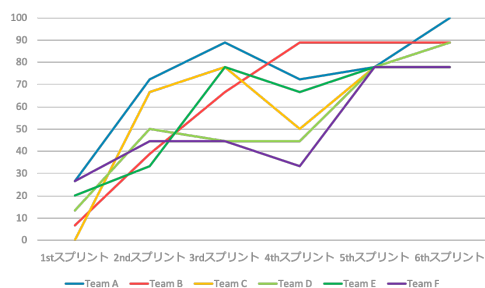


図 8 “Scrum 遵守”領域の評価結果

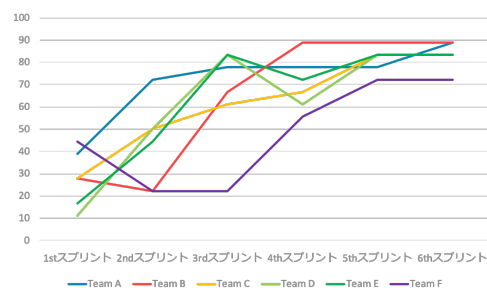


図 11 “検証”領域の評価結果

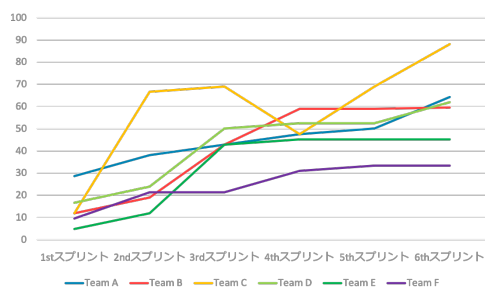


図 9 “プロジェクト計画策定”領域の評価結果

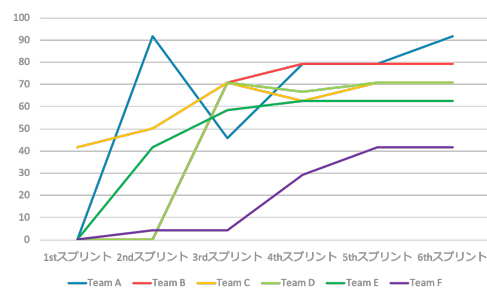


図 12 “測定と分析”領域の評価結果

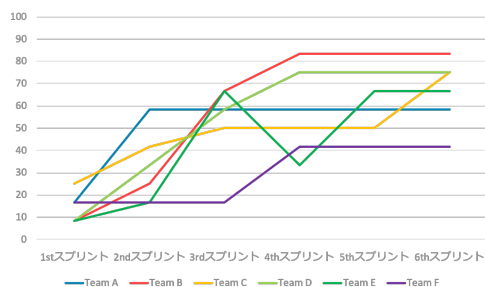


図 10 “プロセスと成果物の品質保証”領域の評価結果

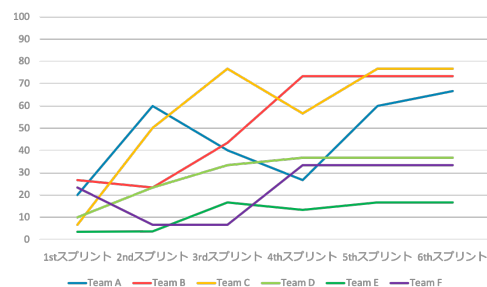


図 13 “プロジェクトの監視と制御”領域の評価結果

は Web API を公開しており、リアルタイムなデータを取得することが可能である。システム構成は図 14 の通りである。

Slack では、User・Channel ごとの投稿数や共有ファイル数、投稿時間などを Web API から取得できる。それらを Power BI をつかって、可視化することで、より分析を容易にしていける。今回は JSON 形式の Slack-log を SQL 形式に変換してくれることから、CData Rest Driver を利用した。

6.2 Slack-log の可視化

FinTech プロジェクトでは、実験対象チーム以外のチームも同 Slack(ワークスペース)に参加していることから、正確な分析が行えないと判断した。一方で、ミライケータイププロジェクトの 3 サービスは、独立した個々の Slack(ワークスペース)を利用しており分析に適していると考え、可視化対象とした。特に評価結果の中で、特徴のある部分の Slack-log を可視化することにより、客観的かつ定量的なデータから評価結果を分析していく。



図 14 Slack の情報可視化システムの構成

7 分析、考察

7.1 全プロセス領域の相加平均値の変動

図 15 と図 16 は、全プロセス領域の相加平均値をプロジェクト/チーム別にスプリントごとに算出したグラフである。

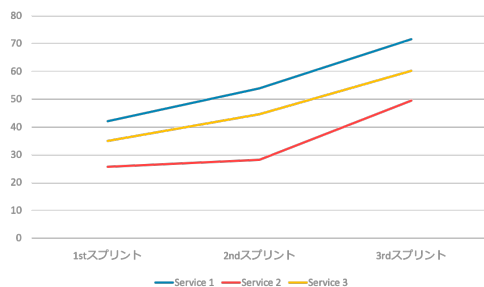


図 15 ミライケータイプロジェクトの相加平均値の推移

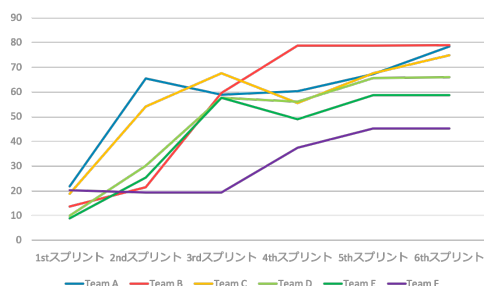


図 16 FinTech プロジェクトの相加平均値の推移

図 15・16 より、改善効果が最も高かったのは、FinTech/TeamB で 65.3%、最も低かったのは、ミライケータイ/Service3 の 23.8%であった。

FinTech の TeamB は、主に学部 1 年生で構成され

たチームであり、開発経験が少ないのにくわえ、これが初めてのチーム開発となる。開発未経験からチーム開発を経験し、自分たちを客観的に分析する能力が向上したことや、周りの上級生のチームから良いところを吸収する力が高かったことから、数値の変動が大きかったと推測できる。また、同プロジェクト内に学部 4 年生以上のチームメンバーがいたこと、スプリントの回数がミライケータイプロジェクトよりも多かったことも数値が高い要因であると考えられる。

ミライケータイ/Service3 は、主な開発が実験期間外で行われていたことや、スプリント数が少なく、改善のタイミングが少なかったことが改善効果が低くなってしまった要因であると考えられる。

7.2 各プロセス領域の数値変動

プロセス領域ごとの評価データである図 2～図 13 をみていくと、プロジェクトごとに改善効果・評価数値の高い領域が異なっていることがわかった。

7.2.1 ミライケータイプロジェクト

ミライケータイプロジェクトでは、3 サービスの平均が最も高い改善効果・評価数値を示したのはプロジェクトの監視と制御、最も高い評価結果となったのは、Service3 のプロジェクト計画策定だった。これは、遠隔で開発を行うプロジェクトであることから、遠隔間の進捗や状況の把握をより求め、成熟度を向上させたことが要因であると考えられる。

マネジメントに関する領域が重点的に改善されており、マネジメントを意識したプロセスを形成していたことがわかった。

7.2.2 FinTech プロジェクト

FinTech プロジェクトでは、Scrum 遵守の領域が最も高い改善効果・評価数値を示したプロセス領域だった。FinTech プロジェクトでは、1 スプリント辺りの期間を短くすることで、ミライケータイプロジェクトよりも Scrum イベントや改善活動を多く実施していた。そのため、プロジェクトとして、Scrum Guide に記載されたルールやイベントに対する意識が高まっていたことから、改善効果・評価数値が高かったと考えられる。

7.3 評価データと可視化情報

図 17 は、ミライケータイプロジェクト/Service2 の各プロセス領域の評価結果をスプリントごとにグラフ化したものである。縦軸は評価結果(%), 横軸はスプリントを表している。領域の A~F は、表 1 と対応しており、領域 C/D が全て同値だったため、合わせて表記している。

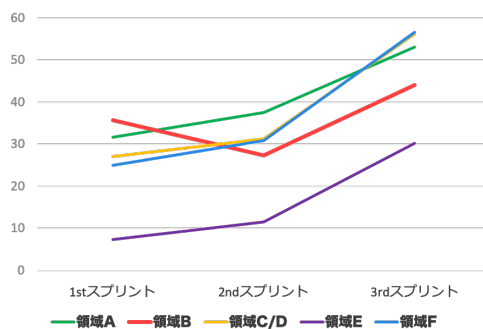


図 17 Service2-各プロセス領域の評価結果

Service2 の領域 B は、ミライケータイプロジェクトのうち、唯一評価結果が一度下降し、再度上昇しているプロセス領域である。領域 B は、プロジェクトの監視と制御の領域であり、2nd スプリントで下降し、3rd スプリントで再度数値が上昇していることがわかる。

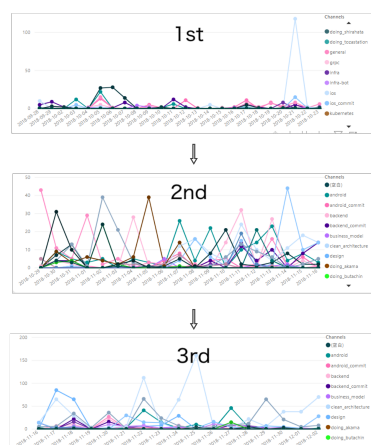


図 18 Service2-Channels ごとのメッセージ数の推移

図 18 は、Service2 の Slack-log から、Channels ごとのメッセージ数の推移について、日付ごとに可視化したものである。縦軸は発言 (投稿) 数、横軸は日付を表している。1st スプリントと 2nd スプリントを比較すると、チャンネル数が急激に増加していることがわかる。また、2nd スプリントと 3rd スプリントを比較すると、チャンネル数が落ち着き、いくつかのチャンネルで活発なメッセージのやりとりをしていることがわかる。

2nd スプリントでチャンネル作成のルールや区別などを決めずに急激に Channel 数が増加したことで、統制が取れず透明性が下がり、全体の状況把握が困難だった可能性が高い。これらの理由から、プロジェクトの監視と制御の評価結果に変動がみられたことがわかった。

Slack などの Communication Tool を可視化することにより、評価データや報告書などからはわからないプロセス形成、学生によるアセスメント効果についての分析が可能であることがわかった。

8 おわりに

本研究では、アジャイルソフトウェア開発 PBL を実践する学生の学び・Scrum 導入の支援および講義指導、カリキュラム改善などの複数の課題解決を目的とした定量的な評価手法を提案し、実際に行われている PBL に適用した。結果として、学生自身がプロジェクトの状況を検査することができ、それらを活用することでスプリントごとに、形成的なアセスメントを学生自身で行うことができた。また、評価データを Communication Tool の可視化情報と合わせて分析することで、PBL の形成過程であるプロセスを評価できることがわかった。

今後は、教育者からみた形成的アセスメントの効果について、アンケートやヒアリングなどから定性的に検証していく。

参考文献

- [1] 文部科学省:「成長分野を支える情報技術人材の育成拠点の形成(enPiT)」公募要領, pp. 2, 2017.
- [2] Ken Schwaber, Jeff Sutherland: Scrum Guide, 2017, <https://www.scrumguides.org/docs/scrumguide/v2017/>, <cited:2019.07.28 >.
- [3] 文部科学省: enPiT 成長分野を支える情報技術人材の育成拠点の形成, <http://www.enpit.jp/>, <cited:2019.07.28 >.
- [4] Rachel Davies(著), Liz Sedley(著), 永瀬美穂(訳), 角征典(訳): アジャイルコーチング, オーム社, 2017.
- [5] 大隅智春, 鴻巣努, 関哲朗, 新井浩志, 西尾雅年: プロジェクトベース教育の効果に関する考察, プロジェクトマネジメント学科研究発表大会予稿集, pp. 179 - 180, 1999.
- [6] 本庄加代子: PBL の課題克服に向けたプロジェクトマネジメント理論の有効性, 東洋学園大学紀要, 25 号, 145-164, 2017.
- [7] 櫻井浩子, 山本雅基, 海上智昭, 小林隆志, 宮地充子, 奥野拓, 桑野文洋, 春名修介, 井上克郎: 情報系大学院生に対する実践教育の効果測定, 65 巻, 1 号, pp.52-57, 2017.
- [8] CMU/SEI: CMMI for Development-Version 1.3, Vol.1.3, 2010.
- [9] CMU/SEI and 日本 SPI コンソーシアム CMMI V1.3 翻訳研究会: 開発のための CMMI 1.3 版, Vol. 1.3, 2010.
- [10] CMMI Institute: Introducing CMMI V2.0, 2018.
- [11] 美馬のゆり: 未来を創る「プロジェクト学習」のデザイン, 公立はこだて未来大学出版, 2018.
- [12] Slack: Slack, <https://slack.com/>, <cited:2019.08.02 >.
- [13] Microsoft: Power BI, <https://powerbi.microsoft.com/>, <cited:2019.08.02 >.
- [14] CData Software Japan: CData Rest, <https://www.cdata.com/jp/drivers/rest/>, <cited:2019.08.02 >.